CLAIMS

1. In a system having a transmitter transmitting a plurality of packets each containing a header to a receiver, a method of synchronizing the transmission of compressed headers between the transmitter and receiver comprising:

transmitting a current packet from the transmitter to the receiver containing information that the transmitter is prepared to send subsequently transmitted packets in which the headers therein are to be compressed in comparison to the header contained in the current packet; and

transmitting from the receiver to the transmitter an acknowledgment packet that the receiver has received the current packet.

2. A method in accordance with claim 1 wherein:

the transmitter stores the header of the current packet which has been acknowledged as being received by the receiver as a reference header which is used in the transmission of the subsequently transmitted packets as a reference header to be used by the receiver to decompress the subsequent headers;

the receiver stores the header of the current packet, which is acknowledged, for decompressing the compressed headers of the subsequently transmitted packets;

the transmitter transmits the subsequent packets using the stored header of the current packet as a reference header; and

the receiver uses the stored referenced header to decompress the compressed headers of the subsequently transmitted received packets to produce a full header which is not compressed.

3. A method in accordance with claim 1 wherein:

the header of the current packet is a full header; and

the compressed header of the subsequently transmitted packets is a first order compressed header.

4. A method in accordance with claim 2 wherein:

the header of the current packet is a full header; and

the compressed header of the subsequently transmitted packets is a first order compressed header.

5. A method in accordance with claim 1 wherein:

the header of the current packet is a first order compressed header; and

the compressed header of the subsequently transmitted packets is a second order compressed header.

6. A method in accordance with claim 2 wherein:

the header of the current packet is a first order compressed header; and

the compressed header of the subsequently transmitted packets is a second order compressed header.

7. A method in accordance with claim 1 wherein:

the header of the current packet is a full header; and

the compressed header of the subsequently transmitted packets is a second order compressed header.

8. A method in accordance with claim 2 wherein:

the header of the current packet is a full header; and

the compressed header of the subsequently transmitted packets is a second order compressed header.

9. A system comprising:

a transmitter which transmits a plurality of packets each containing a header;

a receiver which receives the transmitted plurality of packets; and wherein

the transmitter transmits a current packet to the receiver containing information that the transmitter is prepared to send subsequently transmitted

packets in which the headers therein are to be compressed in comparison to the current packet and the receiver transmits an acknowledgment packet that the receiver has received the current packet.

10. A system in accordance with claim 9 wherein:

the transmitter stores the header of the current packet, which has been acknowledged as being received by the receiver, as a reference header that is used in the transmission of the subsequently transmitted packets as a reference header to be used by the receiver to decompress the subsequent headers;

the receiver stores the header of the current packet which is acknowledged as a reference header for decompressing the compressed headers of the subsequently transmitted packets;

the transmitter transmits the subsequent packets using the stored header of the current packet as a reference header; and

the receiver uses the stored reference header to decompress the compressed headers of the subsequently transmitted received packets to produce a full header which is not compressed.

11. A system in accordance with claim 9 wherein:

the header of the current packet is a full header; and

the compressed header of the subsequently transmitted packets is a first order compressed header.

12. A system in accordance with claim 10 wherein:

the header of the current packet is a full header; and

the compressed header of the subsequently transmitted packets is a first order compressed header.

13. A system in accordance with claim 9 wherein:

the header of the current packet is a first order compressed header; and

the compressed header of the subsequently transmitted packets is a second order compressed header.

14. A system in accordance with claim 10 wherein:

the header of the current packet is a first order compressed header; and

the compressed header of the subsequently transmitted packets is a second order compressed header.

15. A system in accordance with claim 9 wherein:

the header of the current packet is a full header; and

the compressed header of the subsequently transmitted packets is a second order compressed header.

16. A system in accordance with claim 10 wherein:

the header of the current packet is a full header; and

the compressed header of the subsequently transmitted packets is a second order compressed header.

17. In a system having a transmitter transmitting a plurality of packets each containing a header to a receiver, a method of decompressing a compressed header contained in a packet currently received by the receiver comprising:

determining with a counter at the receiver elapsed time Δt between consecutively received packets;

comparing the elapsed time Δt between the currently received packet and an immediately previously received packet to determine if the elapsed time Δt is at least equal to a time lapse indicating that at least one packet is missing between the currently received packet and the immediately previously received packet;

processing the elapsed time Δt indicating that at least one packet is missing to determine a number of missing packets between the immediately previously received packet and the currently received packet;

adding the number of missing packets to a packet number of the immediately previously received packet to update a number of the current packet in a sequence of transmission the plurality of packets; and

decompressing the compressed header of the current packet using the updated number of the current packet.

18. A method in accordance with claim 17 wherein:

the header of the current packet is a second order compressed header.

19. A method in accordance with claim 17 wherein:

a number of packets missing between the immediately previously received packet and the currently received packet is calculated as

$i*SEQ\_CYCLE+DIFF\ L(n_2, n_1)$

if $(DIFF(n_2, n_1) + i*SEQ\_CYCLE)*$

(t time units) $\leq \Delta t < (DIFF(n_2, n_1), +$

$(i+1)*SEQ\_CYCLE*$(t time units)

wherein i is a whole number equal to or greater than zero, n_2 is a sequence number in a sequence of transmission of the packets including the current packet, n_1 is a sequence number in the sequence of transmission of the packets including the immediately previously received packet, SEQ_CYCLE is equal to 2^k wherein k is the number of bits of the sequence number, $DIFF(n_2, n_1)$ is the difference in packet number between in the current and immediately previously received packets and $DIFF(n_2, n_1)=n_2-n_1$ when $n_2>n_1$ and $DIFF(n_2, n_1) = n_2+2^L-n_1$ when $n_2 \leq n_1$.

20. A method in accordance with claim 18 wherein:

a number of packets missing between the immediately previously received packet and the currently received packet is calculated as

$i*SEQ\_CYCLE+DIFF\ L(n_2, n_1)$

if $(DIFF(n_2, n_1) + i*SEQ\_CYCLE)*$

(t time units) $\leq \Delta t < (DIFF(n_2, n_1), +$

$(i+1)*SEQ\_CYCLE*$(t time units)

wherein i is a whole number equal to or greater than zero, n_2 is a sequence number in a sequence of transmission of the packets including the current packet, n_1 is a sequence number in the sequence of transmission of the packets including the immediately previously received packet, SEQ_CYCLE is equal to 2^k wherein k is the number of bits of the sequence number, $DIFF(n_2, n_1)$ is the difference in packet number between in the current and immediately previously received packets and $DIFF(n_2, n_1)=n_2-n_1$ when $n_2>n_1$ and $DIFF(n_2, n_1) = n_2+2^k-n_1$ when $n_2 \leq n_1$.

21. A system comprising:

a transmitter transmitting a plurality of packets each containing a header; and

a receiver which receives the transmitted plurality of packets; and wherein

the receiver comprises a counter which determines elapsed time Δt between consecutively received packets, and compares the elapsed time Δt between the currently received packet and an immediately previously received packet to determine if the elapsed time Δt is at least equal to a time lapse indicating that at least one packet is missing between the currently received packet and the immediately previously received packet, processes the elapsed time Δt indicating that at least one packet is missing to determine a number of missing packets between the immediately previously received packet and the currently received packet, adds the number of missing packets to a packet number of the immediately previously received packet to update a number of the current packet in a sequence of transmission the plurality of packets, and decompresses the compressed header of the current packet using the updated number of the current packet.

22. A system in accordance with claim 21 wherein:

the header of the current packet is a second order compressed header.

23. A system in accordance with claim 21 wherein:

a number of packets missing between the immediately previously received packet and the currently received packet is calculated as

$i*SEQ\_CYCLE+DIFF\ L(n_2, n_{1)}$

if $(DIFF(n_2, n_1) + i*SEQ\_CYCLE)*$

(t time units) $\leq \Delta t <$ $(DIFF(n_2, n_1), +$

$(i+1)*SEQ\_CYCLE*$(t time units)

wherein i is a whole number equal to or greater than zero, n_2 is a sequence number in a sequence of transmission of the packets including the current packet, n_1 is a sequence number in the sequence of transmission of the packets including the immediately previously received packet, SEQ_CYCLE is equal to 2^k wherein k is the number of bits of the sequence number, $DIFF(n_2, n_1)$ is the difference in packet

number between in the current and immediately previously received packets and DIFF(n_2, n_1)=n_2-n_1 when n_2>n_1 and DIFF(n_2, n_1) = n_2+2^k-n_1 when $n_2 \leq n_1$.

24. A system in accordance with claim 22 wherein:

a number of packets missing between the immediately previously received packet and the currently received packet is calculated as

i*SEQ_CYCLE+DIFF L(n_2, n_1)

if (DIFF(n_2, n_1) + i*SEQ_CYCLE)*

(t time units) $\leq \Delta t <$ (DIFF(n_2, n_1), +

(i+1)*SEQ_CYCLE*(t time units)

wherein i is a whole number equal to or greater than zero, n_2 is a sequence number in a sequence of transmission of the packets including the current packet, n_1 is a sequence number in the sequence of transmission of the packets including the immediately previously received packet, SEQ_CYCLE is equal to 2^k wherein k is the number of bits of the sequence number, DIFF(n_2, n_1) is the difference in packet number between in the current and immediately previously received packets and DIFF(n_2, n_1)=n_2-n_1 when n_2>n_1 and DIFF(n_2, n_1) = n_2+2^k-n_1 when $n_2 \leq n_1$.

25. In a system having a transmitter transmitting a plurality of packets each containing a header to a receiver, a method of synchronizing transmission of first order compressed headers between the transmitter and receiver comprising:

transmitting a current packet to the receiver containing a first order compression header with a number of the current packet in the plurality of packets being coded by an extended sequence number having ℓ bits;

in response to reception of the current packet containing the first order header, transmitting from the receiver to the transmitter an acknowledgment packet that the receiver has received the current packet containing the first order compressed header; and

in response to reception of the acknowledgment packet, the transmitter transmits subsequent packets each containing a sequence number having non-extended sequence number having K bits with ℓ>k.

26. A method in accordance with claim 25 wherein:

the transmitter stores the header of the current packet, which has been acknowledged as being received by the receiver, as a reference header that is used in the transmission of the subsequently transmitted packets containing a first order compressed header as a reference header to be used by the receiver to decompress the subsequent headers;

the receiver stores the header of the current packet, which is acknowledged as a reference header, for decompressing the compressed headers of the subsequently transmitted packets containing a first order compressed header;

the transmitter transmits subsequent packets containing the first order compressed header using the stored header of the current packet as a reference header; and

the receiver decompresses the compressed headers of the subsequently transmitted received packets containing the first order compressed header with the stored reference header to produce a full header which is not compressed.

27. A method in accordance with claim 25 wherein:

the receiver detects at least one lost packet in the subsequently transmitted packets by comparison of the sequence numbers of successively received transmitted packets.

28. A method in accordance with claim 26 wherein:

the receiver detects at least one lost packet in the subsequently transmitted packets by comparison of the sequence numbers of successively received transmitted packets.

29. A method in accordance with claim 26 wherein:

a number of missing packets is determined between an immediately previously received packet and the current packet;

the number of determined missing packets is added to a packet number of the immediately previously received packet to a number of the current packet to update a number of the current packet in a sequence of transmission of the plurality of packets; and

decompressing a sequence number of the current packet using the updated number and decompressing additional fields of information using the stored reference header.

30. A method in accordance with claim 28 wherein:

a number of missing packets is determined between an immediately previously received packet and the current packet;

the number of determined missing packets is added to a packet number of the immediately previously received packet to a number of the current packet to update a number of the current packet in a sequence of transmission of the plurality of packets; and

decompressing a sequence number of the current packet using the updated number and decompressing additional fields of information using the stored reference header.

31. In a system having a transmitter transmitting a plurality of packets each containing a header to a receiver, a method of synchronizing transmission of first order compressed headers between the transmitter and receiver comprising:

transmitting a plurality of packets to the receiver each containing a first order compressed header with a number of each of the plurality of packets in an order of transmission being defined by a sequence number having ℓ extended bits; and

detecting at least one lost packet in the transmitted plurality of packets between a current packet and a last packet when a difference DIFF equals DIFF (CD_SN_CURR, CD_SN_LAST) wherein CD_SN_LAST is an absolute packet number of a last received packet and CD_SN_CURR is an absolute packet number of the current packet.

32. A method in accordance with claim 31 wherein:

a number of lost packets N_{loss} is calculated to be equal to DIFF EXT(CD_SN_CURR, CD_SN_LAST) is equal to (CD_SN_CURR)-(CD_SN_LAST).

33. A system comprising:

a transmitter which transmits a plurality of packets each containing a header; and

a receiver which receives the plurality of packets each containing a header; and wherein

a current packet is transmitted by the transmitter to the receiver containing a first order compression header with a number of the plurality of packets being coded by a multiple bit sequence number, in response to reception of the current packet containing the first order header the receiver transmits to the transmitter an acknowledgment packet that the receiver has received the current packet containing the first order compressed header and the transmitter in response to reception of the acknowledgment packet transmits subsequent packets each containing a sequence number in the plurality of packets having a reduced number of bits compared to a number of bits in the sequence number of the current packet.

34. A system in accordance with claim 33 wherein:

the transmitter stores the header of the current packet, which has been acknowledged as being received by the receiver, as a reference header that is used in the transmission of the subsequently transmitted packets containing a first order compressed header as a reference header to be used by the receiver to decompress the subsequent headers;

the receiver stores the header of the current packet, which is acknowledged as a reference header, for decompressing the compressed headers of the subsequently transmitted packets containing a first order compressed header;

the transmitter transmits subsequent packets containing the first order compressed header using the stored header of the current packet as a reference header; and

the receiver decompresses the compressed headers of the subsequently transmitted received packets containing the first order compressed header with the stored reference header to produce a full header which is not compressed.

35. A system in accordance with claim 33 wherein:

the receiver detects at least one lost packet in the subsequently transmitted packets by comparison of the sequence numbers of successively received transmitted packets; and

the header of a packet received immediately after a last in time lost packet is decompressed using a detected number of lost packets.

36. A system in accordance with claim 34 wherein:

the receiver detects at least one lost packet in the subsequently transmitted packets by comparison of the sequence numbers of successively received transmitted packets; and

the header of a packet by the receiver immediately after a last in time lost packet is decompressed using the stored reference header and using a detected number of lost packets.

37. A system in accordance with claim 34 wherein:

a number of missing packets is determined by the receiver between an immediately previously received packet and the current packet;

the number of determined missing packets is added by the receiver to a packet number of the immediately received packet to a number of the current packet to update a number of the current packet in a sequence of transmission of the plurality of packets; and

the receiver decompresses a sequence number of the current packet using the updated number and additional fields of information using the stored reference header.

38. A system in accordance with claim 36 wherein:

a number of missing packets is determined by the receiver between an immediately previously received packet and the current packet;

the number of determined missing packets is added by the receiver to a packet number of the immediately received packet to a number of the current packet to update a number of the current packet in a sequence of transmission of the plurality of packets; and

the receiver decompresses a sequence number of the current packet using the updated number and additional fields of information using the stored reference header.

39. In a system having a transmitter transmitting a plurality of packets each containing a header to a receiver, a method of synchronizing the transmission of headers between the transmitter and receiver comprising:

transmitting from the receiver to the transmitter periodic acknowledgments which are individually transmitted to the transmitter with a spacing such that the transmitter receives an acknowledgment at least once every N packets where $N=2^K$ and k is a number of bits used to number the packets in sequence; and

in an absence of the transmitter receiving a properly timed acknowledgment from the receiver, the receiver increases the number of bits defining the sequence number to be ℓ extended bits wherein ℓ extended is larger than k.

40. A method in accordance with claim 39 wherein:

the receiver can detect a maximum number of lost packets equal $2^{\ell \text{ extended}}$ bits.

41. A method in accordance with claim 39 wherein:

the transmitter, in response to a subsequently received acknowledgment, reduces the number of bits in the sequence numbers from ℓ extended bits to k bits.

42. A method in accordance with claim 40 wherein:

the transmitter, in response to a subsequently received acknowledgment, reduces the number of bits in the sequence numbers from ℓ extended bits to k bits.

43. In a system having a transmitter which transmits a plurality of packets to a receiver, each of the packets containing a header, a method of maintaining sequence synchronization during transmission of packets having compressed headers between the transmitter and the receiver comprising:

initiating transmission of packets having compressed headers by transmitting from the transmitter to the receiver a packet having a full header;

transmitting from the transmitter to the receiver, subsequent to transmission of the packet having the full header, packets having compressed headers, each compressed header containing information related to the full header of the packet having a full header; and

periodically transmitting from the receiver to the transmitter an acknowledgment packet indicating that the packets having the compressed headers have been received.

44. A method according to claim 43, wherein the transmitting comprises:

sequentially adding to the compressed header of each of the packets having compressed headers a sequence number which is incremented by one for each sequential packet of the packets having compressed headers, the sequence number has a predetermined number of bits.

45. A method according to claim 44, further comprising:

when the receiver has not received the acknowledgment packet, extending the number of bits of the sequence number beyond the predetermined number of bits.

46. A method of reducing a number of bits contained in headers of a sequence of transmitted data packets comprising:

transmitting at least one sequence of data packets from a transmitter to a receiver with each sequence containing at least one packet containing a full header followed by at least one packet containing a compressed header having fewer bits than the full header;

in response to one of the data packets received by the receiver containing a full header transmitting from the receiver to the transmitter an

acknowledgment that the receiver has received the one data packet containing the full header; and

in response to the receiving of the acknowledgment by the transmitter, transmitting at least one subsequent data packet from the transmitter to the receiver with a header which is further compressed beyond the compression of the at least one header in the at least one sequence.

47. A method in accordance with claim 46 wherein:

the compressed headers of the at least one sequence are first order compressed headers; and

the compressed header of the at least one subsequent packet is a second order compressed header.

48. A method in accordance with claim 46 wherein:

a plurality of sequences of packets are transmitted.

49. A method in accordance with claim 47 wherein:

a plurality of sequences of packets are transmitted.

50. A method in accordance with claim 46 wherein:

the receiver generates the acknowledgment in response to a first received packet containing a full header.

51. A method in accordance with claim 47 wherein:

the receiver generates the acknowledgment in response to a first received packet containing a full header.

52. A method in accordance with claim 48 wherein:

the receiver generates the acknowledgment in response to a first received packet containing a full header.

53. A method in accordance with claim 49 wherein:

the receiver generates the acknowledgment in response to a first received packet containing a full header.

54. A method in accordance with claim 46 wherein:

the receiver transmits at least one additional acknowledgment to the transmitter in response to reception of the at least one packet containing a compressed header.

55. A method in accordance with claim 54 wherein:

the at least one additional acknowledgment is generated in response to a first packet in the at least one sequence.

56. A method in accordance with claim 47 wherein:

the receiver transmits at least one additional acknowledgment to the transmitter in response to reception of the at least one packet containing a compressed header.

57. A method in accordance with claim 56 wherein:

the at least one additional acknowledgment is generated in response to a first packet in the at least one sequence.

58. A method in accordance with claim 48 wherein:

the receiver transmits at least one additional acknowledgment to the transmitter in response to reception of the at least one packet containing a compressed header.

59. A method in accordance with claim 58 wherein:

the at least one additional acknowledgment is generated in response to a first packet in the at least one sequence.

60. A method in accordance with claim 49 wherein:

the receiver transmits at least one additional acknowledgment to the transmitter in response to reception of the at least one packet containing a compressed header.

61. A method in accordance with claim 60 wherein:

the at least one additional acknowledgment is generated in response to a first packet in the at least one sequence.

62. A method in accordance with claim 50 wherein:

the receiver transmits at least one additional acknowledgment to the transmitter in response to reception of the at least one packet containing a compressed header.

63. A method in accordance with claim 62 wherein:

the at least one additional acknowledgment is generated in response to a first packet in the at least one sequence.

64. A method in accordance with claim 51 wherein:

the receiver transmits at least one additional acknowledgment to the transmitter in response to reception of the at least one packet containing a compressed header.

65. A method in accordance with claim 64 wherein:

the at least one additional acknowledgment is generated in response to a first packet in the at least one sequence.

66. A method in accordance with claim 52 wherein:

the receiver transmits at least one additional acknowledgment to the transmitter in response to reception of the at least one packet containing a compressed header.

67. A method in accordance with claim 66 wherein:

the at least one additional acknowledgment is generated in response to a first packet in the at least one sequence.

68. A method in accordance with claim 53 wherein:

the receiver transmits at least one additional acknowledgment to the transmitter in response to reception of the at least one packet containing a compressed header.

69. A method in accordance with claim 69 wherein:

the at least one additional acknowledgment is generated in response to a first packet in the at least one sequence.

70. A method of reducing a number of bits contained in headers of a sequence of transmitted packets comprising:

transmitting at least one sequence of packets from a transmitter to a receiver with each sequence containing at least one packet containing a first header followed by at least one packet containing a second header which is compressed by having fewer bits than the first header;

in response to one of the packets received by the receiver containing the first header transmitting from the receiver to the transmitter an acknowledgment that the receiver has received the one packet containing the first header; and

in response to the receiving of the acknowledgment by the transmitter, transmitting at least one subsequent packet from the transmitter to the receiver with a third header which is further compressed beyond the compression of the second compressed header.

71. A method in accordance with claim 70 wherein:

the second compressed header is a first order compressed header; and

the third header is a second order compressed header.

72. A method in accordance with claim 70 wherein:

a plurality of sequences of packets are transmitted.

73. A method in accordance with claim 71 wherein:

a plurality of sequences of packets are transmitted.

74. A method in accordance with claim 70 wherein:

the receiver generates the acknowledgment in response to a first received packet containing the first header.

75. A method in accordance with claim 71 wherein:

the receiver generates the acknowledgment in response to a first received packet containing the first header.

76. A method in accordance with claim 72 wherein:

the receiver generates the acknowledgment in response to a first received packet containing the first header.

77. A method in accordance with claim 73 wherein:

the receiver generates the acknowledgment in response to a first received packet containing the first header.

78. A method in accordance with claim 70 wherein:

the receiver transmits at least one additional acknowledgment to the transmitter in response to reception of the at least one packet containing a compressed header.

79. A method in accordance with claim 78 wherein:

the at least one additional acknowledgment is generated in response to a first packet in the at least one sequence.

80. A method in accordance with claim 71 wherein:

the receiver transmits at least one additional acknowledgment to the transmitter in response to reception of the at least one packet containing a compressed header.

81. A method in accordance with claim 80 wherein:

the at least one additional acknowledgment is generated in response to a first packet in the at least one sequence.

82. A method in accordance with claim 72 wherein:

the receiver transmits at least one additional acknowledgment to the transmitter in response to reception of the at least one packet containing a compressed header.

83. A method in accordance with claim 82 wherein:

the at least one additional acknowledgment is generated in response to a first packet in the at least one sequence.

84. A method in accordance with claim 73 wherein:

the receiver transmits at least one additional acknowledgment to the transmitter in response to reception of the at least one packet containing a compressed header.

85. A method in accordance with claim 84 wherein:

the at least one additional acknowledgment is generated in response to a first packet in the at least one sequence.

86. A method in accordance with claim 74 wherein:

the receiver transmits at least one additional acknowledgment to the transmitter in response to reception of the at least one packet containing a compressed header.

87. A method in accordance with claim 86 wherein:

the at least one additional acknowledgment is generated in response to a first packet in the at least one sequence.

88. A method in accordance with claim 75 wherein:

the receiver transmits at least one additional acknowledgment to the transmitter in response to reception of the at least one packet containing a compressed header.

89. A method in accordance with claim 88 wherein:

the at least one additional acknowledgment is generated in response to a first packet in the at least one sequence.

90. A method in accordance with claim 76 wherein:

the receiver transmits at least one additional acknowledgment to the transmitter in response to reception of the at least one packet containing a compressed header.

91. A method in accordance with claim 90 wherein:

the at least one additional acknowledgment is generated in response to a first packet in the at least one sequence.

92. A method in accordance with claim 77 wherein:

the receiver transmits at least one additional acknowledgment to the transmitter in response to reception of the at least one packet containing a compressed header.

93. A method in accordance with claim 92 wherein:

the at least one additional acknowledgment is generated in response to a first packet in the at least one sequence.

94. A method of reducing a number of bits contained in headers of a sequence of transmitted packets comprising:

transmitting at least one sequence of packets from a transmitter to a receiver with each sequence containing at least one packet containing a full header followed by at least one packet containing a compressed header having fewer bits than the full header; and

in response to one of the packets received by the receiver containing a full header transmitting from the receiver to the transmitter an acknowledgment that the receiver has received the one packet containing the full header.

Sub A3

95. A method of reducing a number of bits contained in headers of a sequence of transmitted packets comprising:

transmitting at least one sequence of packets from a transmitter to a receiver with each sequence containing at least one packet containing a first header followed by at least one packet containing a second header which is compressed by having fewer bits than the first header; and

in response to one of the packets received by the receiver containing the first header transmitting from the receiver to the transmitter an acknowledgment that the receiver has received the one packet containing the first header.

96. A method of transmitting packets within a string of packets having compressed headers each containing a sequence number identifying a position of each packet in the string from a transmitter to a receiver comprising:

processing the string to detect when the string contains at least one lost or out of sequence packet prior to transmission of the packet;

transmitting the string with compressed headers from the transmitter to the receiver as a sequence of packets containing packets preceding and succeeding the lost or out of sequence packets; and

transmitting with at least one packet succeeding the at least one lost or out of sequence packet a number of any lost or out of sequence packets in the data string of packets.

97. A method in accordance with claim 96 wherein:

the receiver decompresses at least one succeeding packet which is received including adding the number of any lost or out of sequence packets to the sequence number of each received succeeding packet.

98. A method in accordance with claim 97 wherein:

the decompressing by the receiver uses a stored reference packet which was transmitted as part of the string of packets before the at least one lost or out of sequence packet and includes a sequence number used to decompress the at least one subsequent packet.

99. A method in accordance with claim 96 wherein:
the string contains at least one lost packet.

100. A method in accordance with claim 96 wherein:
the string contains at least one out of sequence packet.

101. A method in accordance with claim 97 wherein:
the string contains at least one lost packet.

102. A method in accordance with claim 97 wherein:
the string contains at least one out of sequence packet.

103. A method in accordance with claim 98 wherein:
the string contains at least one lost packet.

104. A method in accordance with claim 98 wherein:
the string contains at least one out of sequence packet.

105. A method of transmitting packets within a string of packets having compressed headers each containing a sequence number identifying a position of each packet in the string from a transmitter to a receiver comprising:
processing the string to detect when the data string contains at least one lost packet prior to transmission of the packet;
transmitting the string with compressed headers from the transmitter to the receiver as a sequence of packets containing packets preceding and succeeding the lost packets; and
transmitting with at least one packet succeeding the at least one lost packet a number of any lost packets.

106. A method in accordance with claim 105 wherein:
the receiver decompresses at least one succeeding packet which is received including adding the number of any lost packets to the sequence number of each received succeeding packet.

107. A method in accordance with claim 106 wherein:

the decompressing by the receiver uses a stored reference packet which was transmitted as part of the string of packets before the at least one sequence and includes a sequence number used to decompress the at least one subsequent packet.

108. A method of transmitting packets within a data string of packets having compressed headers each containing a sequence number identifying a position of each packet in the string from a transmitter to a receiver comprising:

processing the string to detect when the string contains at least one lost or out of sequence packet prior to transmission of the packet;

transmitting the string with compressed headers from the transmitter to the receiver as a sequence of packets containing packets preceding and succeeding the lost or out of sequence packets; and

transmitting with at least one packet succeeding the at least one out of sequence packet a number of any out of sequence packets in the data string.

109. A method in accordance with claim 108 wherein:

the receiver decompresses at least one succeeding packet which is received including adding the number of any out of sequence packets to the sequence number of each received succeeding packet.

110. A method in accordance with claim 109 wherein:

the decompressing by the receiver uses a stored reference packet which was transmitted as part of the string of packets before the at least one sequence and includes a sequence number used to decompress the at least one subsequent packet.

111. A method of transmitting a string of packets comprising:

processing the string of packets with an error detection process to identify any packets in the string of packets which contain errors;

removing the packets from the string which contain the detected errors; and

transmitting from a transmitter to a receiver the string without the packets which have been removed.

112. A method in accordance with claim 111 wherein:

the error detection process utilizes an error detection code within each packet to identify any packets in the string of packets which contain errors.

113. A method in accordance with claim 112 wherein:

the error detection process processes data in each packet to compute an error detection code and determines if a stored error detection code in each packet matches the computed error detection code and if a match is not found removes the packet from the string and which is removed.

114. A method in accordance with claim 111 wherein:

headers of at least some of the packets are compressed prior to transmission.

115. A method in accordance with claim 114 wherein:

the compression of headers of at least some of the packets occurs after the removal of packets containing errors.

116. A method in accordance with claim 112 wherein:

headers of at least some of the packets are compressed prior to transmission.

117. A method in accordance with claim 116 wherein:

the compression of headers of at least some of the packets occurs after the removal of packets containing errors.

118. A method in accordance with claim 113 wherein:

headers of at least some of the packets are compressed prior to transmission.

119. A method in accordance with claim 118 wherein:

the compression of headers of at least some of the packets occurs after the removal of packets containing errors.

120. A method in accordance with claim 111 wherein:

the transmission of the string is over a limited bandwidth transmission medium with the removal of any packets containing an error reducing use of the limited bandwidth during the transmission.

121. A method in accordance with claim 112 wherein:

the transmission of the string is over a limited bandwidth transmission medium with the removal of any packets containing an error reducing use of the limited bandwidth during the transmission.

122. A method in accordance with claim 113 wherein:

the transmission of the string is over a limited bandwidth transmission medium with the removal of any packets containing an error reducing use of the limited bandwidth during the transmission.

123. A method in accordance with claim 114 wherein:

the transmission of the string is over a limited bandwidth transmission medium with the removal of any packets containing an error reducing use of the limited bandwidth during the transmission.

124. A method in accordance with claim 115 wherein:

the transmission of the string is over a limited bandwidth transmission medium with the removal of any packets containing an error reducing use of the limited bandwidth during the transmission.

125. A method in accordance with claim 116 wherein:

the transmission of the string is over a limited bandwidth transmission medium with the removal of any packets containing an error reducing use of the limited bandwidth during the transmission.

126. A method in accordance with claim 117 wherein:

the transmission of the string is over a limited bandwidth transmission medium with the removal of any packets containing an error reducing use of the limited bandwidth during the transmission.

127. A method in accordance with claim 118 wherein:

the transmission of the string is over a limited bandwidth transmission medium with the removal of any packets containing an error reducing use of the limited bandwidth during the transmission.

128. A method in accordance with claim 119 wherein:

the transmission of the string is over a limited bandwidth transmission medium with the removal of any packets containing an error reducing use of the limited bandwidth during the transmission.

129. A method of decompressing headers of a transmitted string of packets which individually contain a sequence number identifying a position of each transmitted packet in the string of packets prior to transmission comprising:

transmitting the string of packets from a transmitter to a receiver with at least one received sequential packet in the transmitted string of packets not being in the received string of packets;

processing the sequence numbers of the received packets to determine a number of sequential packets of the transmitted string of packets which are not present in the received packets; and

decompressing the header of at least one received packet succeeding the at least one packet not present in the received packets by using the number of sequential packets which are determined to not have been received.

130. A method in accordance with claim 129 wherein:

the decompressing of the header of at least one received packet succeeding the at least one packet not present in the received packets is performed by adding a product of the number and an extrapolation function to the header of a received packet which preceded the at least one received packet not present in the received packets.

131. A method in accordance with claim 129 wherein:

a plurality of transmitted packets are not present in the received packets.

132. A method in accordance with claim 130 wherein:

a plurality of transmitted packets are not present in the received packets.

133. A method in accordance with claim 129 wherein:

at least one compressed header in the at least one received packet succeeding the at least one packet which is not present in the received packets is decompressed by adding a plurality of products of the number and different extrapolation functions to the at least one compressed header in the at least one packet succeeding the at least one packet which is not present in the received packets.

134. A method in accordance with claim 130 wherein:

at least one compressed header in the at least one received packet succeeding the at least one packet which is not present in the received packets is decompressed by adding a plurality of products of the number and different extrapolation functions to the at least one compressed header in the at least one packet succeeding the at least one packet which is not present in the received packets.

135. A method in accordance with claim 131 wherein:

at least one compressed header in the at least one received packet succeeding the at least one packet which is not present in the received packets is decompressed by adding a plurality of products of the number and different extrapolation functions to the at least one compressed header in the at least one packet succeeding the at least one packet which is not present in the received packets.

136. A method in accordance with claim 132 wherein:

at least one compressed header in the at least one received packet succeeding the at least one packet which is not present in the received packets is decompressed by adding a plurality of products of the number and different extrapolation functions to the at least one compressed header in the at least one packet succeeding the at least one packet which is not present in the received packets.

137. A method in accordance with claim 130 wherein:

the extrapolation function varies linearly from packet to packet in a plurality of packets succeeding the at least one packet which is not present in the received packets.

138. A method in accordance with claim 131 wherein:

the extrapolation function varies non-linearly within a plurality of packets succeeding the at least one packet which is not present in the received packets.

139. A method in accordance with claim 131 wherein:

the extrapolation function varies linearly from packet to packet in a plurality of packets succeeding the at least one packet which is not present in the received packets.

140. A method in accordance with claim 132 wherein:

the extrapolation functions vary non-linearly within a plurality of packets succeeding the at least one packet which is not present in the received packets.

141. A method in accordance with claim 132 wherein:

the extrapolation functions vary linearly from packet to packet in a plurality of packets succeeding the at least one packet which is not present in the received packets.

142. A method in accordance with claim 133 wherein:

the extrapolation functions vary linearly from packet to packet in a plurality of packets succeeding the at least one packet which is not present in the received packets.

143. A method in accordance with claim 133 wherein:

the extrapolation functions vary non-linearly within a plurality of packets succeeding the at least one packet which is not present in the received packets.

144. A method in accordance with claim 134 wherein:

the extrapolation functions vary linearly from packet to packet in a plurality of packets succeeding the at least one packet which is not present in the received packets.

145. A method in accordance with claim 134 wherein:

the extrapolation functions vary non-linearly within a plurality of packets succeeding the at least one packet which is not present in the received packets.

146. A method in accordance with claim 130 wherein:

the extrapolation function is a time stamp of the at least one received packet succeeding the at least one packet which is not present in the received packets.

147. A method in accordance with claim 130 wherein:

the extrapolation function represents at least one of the sequence number of the at least one received packet succeeding the at least one packet or an IP ID of the at least one received packet which is not present in the received packets.

148. A method in accordance with claim 132 wherein:

the extrapolation function is a representation of time stamp of the at least one received packet succeeding the at least one packet which is not present in the received packets.

149. A method in accordance with claim 132 wherein:

the extrapolation function represents at least one of the sequence number of the at least one received packet succeeding the at least one packet or an IP ID of the at least one received packet which is not present in the received packets.

150. A method in accordance with claim 133 wherein:

the extrapolation functions are a time stamp and a representation of the sequence number of the at least one received packet succeeding the at least one packet which is not present in the received packets.

151. A method in accordance with claim 134 wherein:

the extrapolation functions are a time stamp and a representation of the sequence number of the at least one received packet succeeding the at least one packet which is not present in the received packets.

152. A method in accordance with claim 135 wherein:

the extrapolation functions are a time stamp and a representation of the sequence number of the at least one received packet succeeding the at least one packet which is not present in the received packets.

153. A method in accordance with claim 136 wherein:

the extrapolation functions are a time stamp and a representation of the sequence number of the at least one received packet succeeding the at least one packet which is not present in the received packets.

154. A method in accordance with claim 137 wherein:

the extrapolation functions are a time stamp and a representation of the sequence number of the at least one received packet succeeding the at least one packet which is not present in the received packets.

155. A method in accordance with claim 129 wherein:

the compressed headers are second order compressed headers.

156. A method in accordance with claim 130 wherein:

the compressed headers are second order compressed headers.

157. A method in accordance with claim 133 wherein:

the compressed headers are second order compressed headers.

158. A method in accordance with claim 134 wherein:

the compressed headers are second order compressed headers.

159. A method of regenerating headers of compressed packets within a string of packets which individually contain a sequence number identifying a position of each transmitted packet in the string of packets comprising:

transmitting the string of packets from a transmitter to a receiver with at least one received packet in a sequence within the transmitted string being received with an erroneous compressed header;

storing the at least one received packet in at least one sequence having a header which is erroneous;

determining a number of packets in each stored sequence; and

when a number of stored packets in at least one sequence matches a number determined by processing the sequence numbers of the packets preceding and succeeding the at least one sequence, regenerating the compressed

headers of at least one stored sequence by adding a function of an extrapolation function to a header of at least one packet of at least one sequence.

160. A method in accordance with claim 159 wherein:

the function of an extrapolation function which is added to a header of a plurality of packets of the at least one sequence increases linearly between sequential packets in the at least one sequence.

161. A method in accordance with claim 159 wherein:

the function of an extrapolation function which is added to a header of a plurality of packets of the at least one sequence increases non-linearly within the packets in the at least one sequence.

162. A method in accordance with claim 159 wherein:

a function of a plurality of different extrapolation functions is added to a header of at least one packet of at least one sequence.

163. A method in accordance with claim 162 wherein:

the function of the plurality of different extrapolation functions which is added to a header of a plurality of packets of at least one sequence increases linearly between sequential packets in the at least one sequence.

164. A method in accordance with claim 162 wherein:

the function of the plurality of different extrapolation functions which are added to a header of a plurality of packets of at least one sequence increases non-linearly within packets in the at least one sequence.

165. A method in accordance with claim 159 wherein:

the number of packets in each stored sequence is determined from a difference between the sequence number of the packets immediately preceding and immediately succeeding the at least one sequence.

166. A method in accordance with claim 160 wherein:

the number of packets in each stored sequence is determined from a difference between the sequence number of the packets immediately preceding and immediately succeeding the at least one sequence.

167. A method in accordance with claim 161 wherein:

the number of packets in each stored sequence is determined from a difference between the sequence number of the packets immediately preceding and immediately succeeding the at least one sequence.

168. A method in accordance with claim 162 wherein:

the number of packets in each stored sequence is determined from a difference between the sequence number of the packets immediately preceding and immediately succeeding the at least one sequence.

169. A method in accordance with claim 163 wherein:

the number of packets in each stored sequence is determined from a difference between the sequence number of the packets immediately preceding and immediately succeeding the at least one sequence.

170. A method in accordance with claim 164 wherein:

the number of packets in each stored sequence is determined from a difference between the sequence number of the packets immediately preceding and immediately succeeding the at least one sequence.

171. A method in accordance with claim 159 wherein:

the extrapolation function is a time stamp.

172. A method in accordance with claim 159 wherein:

the extrapolation function is at least one of a sequence number or an IP ID of the at least one received packet.

173. A method in accordance with claim 160 wherein:

the extrapolation function is a time stamp.

174. A method in accordance with claim 160 wherein:

the extrapolation function is at least one of a sequence number or an IP ID of the at least one received packet.

175. A method in accordance with claim 161 wherein:

the extrapolation function is a time stamp.

176. A method in accordance with claim 161 wherein:

the extrapolation function is at least one of a sequence number or an IP ID of the at least one received packet.

177. A method in accordance with claim 162 wherein:

the extrapolation functions are at least one of a time stamp, an IP ID, or a sequence number.

178. A method in accordance with claim 163 wherein:

the extrapolation functions are at least one of a time stamp, an IP ID, or a sequence number.

179. A method in accordance with claim 164 wherein:

the extrapolation functions are at least one of a time stamp, an IP ID, or a sequence number.

180. A method in accordance with claim 166 wherein:

the extrapolation function is a time stamp.

181. A method in accordance with claim 166 wherein:

the extrapolation function is at least one of a sequence number or an IP ID of the at least one received packet.

182. A method in accordance with claim 167 wherein:

the extrapolation function is a time stamp.

183. A method in accordance with claim 167 wherein:

the extrapolation function is at least one of a sequence number or an IP ID of the at least one received packet.

184. A method in accordance with claim 168 wherein:

the extrapolation functions are a timestamp and a sequence number.

185. A method in accordance with claim 169 wherein:

the extrapolation functions are at least one of a time stamp, an IP ID, or a sequence number.

186. A method in accordance with claim 170 wherein:

the extrapolation functions are at least one of a time stamp, an IP ID, or a sequence number.

187. A method of transmitting headers from a compressor to a decompressor comprising:

transmitting at least one packet from a compressor to a decompressor;

in response to receiving the at least one packet at the decompressor transmitting at least one feedback to the compressor signalling that the decompressor has received the at least one packet; and

in response to the feedback, the compressor transmits at least one additional packet to the decompressor which has a smaller number of bits in a header of the at least one additional packet than a number of bits of a header in the at least one packet.

188. A method in accordance with claim 187 wherein:

each header of the at least one packet is a full header; and

each header of the at least one additional packet is a first order header.

189. A method in accordance with claim 187 wherein:

each header of the at least one packet is a first order header; and

each header of the at least one additional packet is a second order header.

190. A method in accordance with claim 187 wherein:

the feedback is an acknowledgment packet.

191. A method of transmitting headers from a compressor to a decompressor comprising:

transmitting a plurality of packets from a compressor to a decompressor;

in response to receiving the at least one packet at the decompressor, transmitting at least one feedback to the compressor signalling that the decompressor has received at least one of the plurality of packets; and

transmitting at least one additional packet from the compressor to the decompressor which has a smaller number of bits in a header of the at least one additional packet than a number of bits of a header in the at least one packet when whichever first occurs of

(1) a transmission of a predetermined number of packets of the at least one packet, or

(2) reception of the at least one feedback.

192. A method in accordance with claim 191 wherein:

each header of the at least one packet is a full header; and

each header of the at least one additional packet is a first order header.

193. A method in accordance with claim 191 wherein:

each header of the at least one packet is a first order header; and

each header of the at least one additional packet is a second order header.

194. A method in accordance with claim 191 wherein:
the feedback is an acknowledgment packet.

195. A method of transmitting headers from a compressor to a decompressor comprising:
transmitting a plurality of packets from a compressor to a decompressor; and
transmitting at least one additional packet from the compressor to the decompressor which has a smaller number of bits in a header of the at least one additional packet than a number of bits of a header in the at least one packet when a transmission of a predetermined number of packets of the at least one packet has occurred.

196. A method in accordance with claim 195 wherein:
the predetermined number of packets is based upon a selection criteria.

197. A method in accordance with claim 196 wherein:
the selection criteria are based upon channel conditions involving transmissions to the decompressor from the compressor or transmissions from the decompressor to compressor.

198. A method in accordance with claim 191 wherein:
the predetermined number of packets is based upon a selection criteria.

199. A method in accordance with claim 198 wherein:
the selection criteria are based upon channel conditions involving transmissions to the decompressor from the compressor or transmissions from the decompressor to compressor.

200. In a system having a transmitter which transmits a plurality of packets to a receiver, each of the packets containing a header, a method of maintaining sequence synchronization during transmission of packets having compressed headers between the transmitter and the receiver comprising:

initiating transmission of packets having headers by transmitting from the transmitter to the receiver a packet having a header;

transmitting from the transmitter to the receiver, subsequent to transmission of the packet having the header, packets having compressed headers, each compressed header containing information related to the header of the packet; and

nonperiodically transmitting from the receiver to the transmitter an acknowledgment packet indicating that the packets having the compressed headers have been received.